WHAT IS CLAIMED IS:

1. An address lookup structure comprising:

at least one hash table storing prefixes for address lookups; and

a content addressable memory storing at least some prefixes for which a collision occurs within the at least one hash table.

2. The address lookup structure according to claim 1, wherein the at least one hash table is contained within a smallest number of memory blocks sufficient to hold all required prefixes for which no collision occurs within the at least one hash table.

3. The address lookup structure according to claim 1, wherein the at least one hash table is contained within a predetermined limited number of memory blocks.

4. The address lookup structure according to claim 1, wherein the at least one hash table contains prefixes hashed by one of two hash functions, a second of the two hash functions employed when a collision occurs with a first of the two hash functions.

5. The address lookup structure according to claim 1, wherein the at least one hash table comprises a plurality of hash tables, each hash table containing different length prefixes.

6. The address lookup structure according to claim 5, further comprising:

a priority encoder selecting a longest prefix when a plurality of matches occur between different length portions of a prefix and prefixes in each of two or more of the plurality of hash tables.

7. The address lookup structure according to claim 5, wherein the plurality of hash tables contain only a subset of different length prefixes possible under an addressing scheme, and wherein a remainder of the different length prefixes are stored in the content addressable memory.

8. A network router including the address lookup structure according to claim 1, the network router further comprising:

a network search engine containing the at least one hash table and coupled to the content addressable memory, the network search engine performing address lookups using the at least one hash table; and

an external memory coupled to the network search engine and containing per route information indexed by a next hop index generated by the network search engine.

9. A network including a plurality of interconnected network routers according to claim 8.

10. An address lookup structure comprising:

a plurality of hash tables each containing prefixes of a different length than prefixes within other hash tables within the plurality, the hash tables collectively containing only a subset of different prefix lengths less than or equal to an address length; and

an additional address lookup facility handling a remainder of the different address lengths not accommodated by the plurality of hash tables.

11. The address lookup structure according to claim 10, wherein the additional address lookup facility comprises a content addressable memory.

12. The address lookup structure according to claim 10, wherein each of the plurality of hash tables in contained in one or more memory blocks allocated based on hashing of each prefix contained in the respective hash table using at least a first hash function,

wherein a number of memory blocks allocated to the respective hash table does not exceed a predefined number, and

wherein a remainder of prefixes of a length corresponding to prefixes within the respective hash table are handled by the additional address lookup facility.

13. The address lookup structure according to claim 10, further comprising:

a priority encoder selecting a longest prefix match from matches identified within the plurality of hash tables.

14. A method of operating an address lookup comprising:

storing at least some address prefixes in at least one hash table; and

storing address prefixes for which a collision occurs within the at least one hash table in a content addressable memory.

15. The method according to claim 14, further comprising:

maintaining the at least one hash table within a smallest number of memory blocks sufficient to hold all required prefixes for which no collision occurs within the at least one hash table.

16. The method according to claim 14, further comprising:

maintaining the at least one hash table within a predetermined limited number of memory blocks.

17. The method according to claim 14, further comprising:

hashing prefixes in the at least one hash table with one of two hash functions, a second of the two hash functions employed when a collision occurs with a first of the two hash functions.

18. The method according to claim 14, further comprising:

storing, in each of a plurality of hash tables, prefixes of a different length than prefixes contained in any other of the plurality of hash tables.

19. The method according to claim 18, further comprising:

selecting a longest prefix when a plurality of matches occur between different length portions of a prefix and prefixes in each of two or more of the plurality of hash tables.

20. The method according to claim 18, further comprising:

storing prefixes corresponding to only a subset of different prefix lengths possible under an addressing scheme in the plurality of hash tables;, and

storing a remainder of prefixes in the content addressable memory.